Project 2 Summary

Dataset 1: Auto MPG

The AutoMPG dataset was a given dataset in this project. It was taken from the ScalaTion code base. The shape of the original dataset was 392x8. However, the origin column was dropped because it was categorical. The data was also checked for any columns that had perfect collinearity. While cylinders, displacement, horsepower, and weight had fairly high collinearity, no columns in the dataset were perfectly collinear, so the final input features were the following: cylinders, displacement, horsepower, weight, acceleration, and model_year. The target value in this case was mpg.

After testing many different models on the AutoMPG dataset, we found that a 4L neural network with a Tanh activation function seemed to yield the best results in Scala. It functioned best not only when pure cross validation was used to produce an R^2 bar of 0.896, a R^2 cv of 0.898, AIC of -164.52, and BIC of -145.66, but also performed well when used with forward selection producing an R^2 bar of 0.877, a R^2 cv of 0.882, an AIC of 181.76, and a BIC of 167.62 for the following features: weight, model year, displacement, and horsepower.

In Python, however, we found different results. When using pure cross validation, we found that a 4L neural network with a reLU activation function performed best with an R^2 bar of 0.836, a R^2 cv of 0.838, AIC of -378.37, and BIC of -364.20. For the forward selection approach, we found that a 3L neural network with reLU activation worked best with 2 input features: model year and weight. This combination yielded an R^2 bar of 0.835, a R^2 cv of 0.835, an AIC of -386.22, and a BIC of -381.49.

Overall, the results showed that Scala out performed Python in this case with a 4L neural network with a Tanh activation function. To view the specific hyperparameters associated with the models or the full set of results please refer to the results.xlsx file that is located in the same directory as the current file.

Dataset 2: Forest Fires

The Forest Fires dataset was a given dataset in this project. It was taken from the UCI Machine Learning Repository. The shape of the original dataset was 517x13. This dataset had a large number of 0 values in the target column area. In an effort to smooth the data and make it more linear, we performed a log transform on the area column making a new column called log_area. We then dropped the original area column so the data retained its original shape of 517x13. The data was also checked for any columns that had perfect collinearity. No columns in the dataset were perfectly collinear and there was low collinearity in general between the columns. The final input features were the following: X, Y, month, day, FFMC, DMC, DC, ISI, temp, RH, wind, and rain. The target value in this case was log_area.

With tests on different models on the ForestFires dataset, we found that a 3L neural network with a reLU activation function seemed to yield the best results for cross validation and 2L with a reLU activation function gave the best results for forward selection in Scala. For cross validation it produced an R^2 bar of -0.005, a R^2 cv of -0.002, AIC of -150.068, and BIC of -113.18. For forward selection it produced an R^2 bar of -0.0409, a R^2 cv of -0.0329, AIC of -188.472, and BIC of -175.298 for the following features: X, Y, Day, rain.

For python tests on the ForestFires dataset, we found that a 4L neural network with a Sigmoid activation function seemed to yield the best results for cross validation and 4L with a Tanh activation function gave the best results for forward selection in Python. For cross validation it produced an R^2 bar of -0.03, a R^2 cv of -0.008, AIC of -309.299, and BIC of -277.636. For forward selection it produced an R^2 bar of 0.006, a R^2 cv of 0.006, AIC of -332.784, and BIC of -330.145 with month being the only feature applied.

Overall, none of the results were very good for this dataset. The best results came from Python with a 4L neural net and Tanh activation function, but an R^2 bar of 0.006 is still far from ideal. To view the specific hyperparameters associated with the models or the full set of results please refer to the results.xlsx file that is located in the same directory as the current file.

Dataset 3: CCPP

The CCPP dataset was a retrieved dataset in this project. It was taken from the UCI Machine Learning Repository. The shape of the original dataset was 9568x5. There was no preprocessing necessary for this because there were no missing/null values, nor any non-ordinal/numeric values. The data was also checked for any columns that had perfect collinearity. No columns in the dataset were perfectly collinear. The input features for the dataset were AT (temperature), V (exhaust vacuum), AP (ambient pressure), and RH(relative humidity). The target variable was PE (power plant electrical output).

For our test on different models on the CCPP dataset, we found that a 4L neural network with a Sigmoid activation function seemed to yield the best results in Scala. It functioned best not only when pure cross validation was used to produce an R^2 bar of 0.938, a R^2 cv of 0.938, AIC of -5472.757, and BIC of -5439.318, but also performed well when used with forward selection producing an R^2 bar of 0.944, a R^2 cv of 0.944, an AIC of -5357.330, and a BIC of -5323.990 for the following features: AT, V, RH, PE.

For python tests on the ForestFires dataset, we found that a 2L neural network with a Tanh activation function seemed to yield the best results for cross validation and 4L with a Tanh activation function gave the best results for forward selection in Python. For cross validation it produced an R^2 bar of 0.936, a R^2 cv of -0.936, AIC of -10956.951, and BIC of -10934.725. For forward selection it produced an R^2 bar of 0.937, a R^2 cv of 0.937, AIC of 10972.736, and BIC of 10950.510 for the following features: AT, RH, V, AP.

Overall, the results showed Scala and Python performed very closely with one another. Scala slightly outperformed Python when operating with a 4L neural network with a Sigmoid activation function. To view the specific hyperparameters associated with the models or the full set of results please refer to the results.xlsx file that is located in the same directory as the current file.

Dataset 4: Bike Sharing

The Bike Sharing dataset was a chosen dataset in this project. It was taken from the UCI Machine Learning Repository. The shape of the original dataset was 17379x15. Due to no missing or null values, nor any non-ordinal/numeric values, no preprocessing was necessary. There were 2 variables with relatively high collinearity, such as month and season, and atemp (perceived temperature) and temp (actual temperature) were almost perfectly collinear. As such, atemp was dropped. The target value in this case was cnt, or the count of total rental bikes that hour.

From the results that we have gathered for the Bike Sharing dataset, the 4L neural network with Tanh activation function produced the best results in Scala. Using cross-validation, we got an R^2 bar of 0.334, a R^2 cv of 0.335, AIC of -1043.785, and BIC of -980.260. However, different results were yielded using forward selection. The 2L neural network with the Tanh activation function provided the best results for this particular dataset. It gave a R^2 bar of 0.293, a R^2 cv of 0.294, AIC of 18.241, and BIC of 42.673 for the following features: season, month, holiday, windspeed.

Switching over to python, the 4L neural network with the reLU activation function gave the best results with cross-validation. For this model, we achieved a R^2 bar of 0.900, a R^2 cv of 0.900, AIC of -19685.846, and BIC of -19618.154. Not only did this specific model work with cross-validation, it also worked with forward selection. The forward selection gave us a R^2 bar of 0.903, a R^2 cv of 0.903, AIC of -19804.951, and BIC of -19737.262 for the following features: hr, atemp, yr, hum, season, windspeed, weekday, holiday, weathersit, working day, and mnth.

Overall, it seemed like two different languages provided different results. Despite the differences in the result, Python provided the best results for this particular dataset with the 4L neural network using the reLU activation function. To view more of the values that we collected from different models, refer to our results excel sheet.

Dataset 5: Wine Quality

Within this wine quality dataset, there are two datasets included, related to red and white wine samples, from the north of Portugal. The goal is to model wine quality based on physicochemical tests using a variety of regression models. The dataset is pulled from UCI Machine Learning Repository. There are 11 inputs (fixed acidity, volatile acidity, citric acid, etc.) and the output is based on sensory data made by wine experts. The quality ranges from 0 (very bad) to 10 (very excellent). Due to the fact that white wine had more instances than red wine, we decided to go with the white wine data set (4898 instances). There are no missing attribute values.

After running through all the models with cross validation in Scala, the 4L neural network with the Tanh activation function yielded the best results for the wine quality dataset. The model gave a R^2 bar of 0.326, a R^2 cv of 0.328, AIC of -1049.119, and BIC of -985.594. However, with forward selection, the 2L neural network with Tanh activation function provided the best results. It gave a R^2 bar of 0.293, a R^2 cv of 0.294, AIC of 18.248, and BIC of 42.680 for the following features: fixed acidity, citric acid, chlorides, and quality.

In Python, we found that the 3L neural network with reLU activation function and cross-validation yielded the best results for the wine quality dataset. It gave a R^2 bar of 0.348, a R^2 cv of 0.348, AIC of -4146.806, and BIC of -4093.048. Not only did the model work well with cross-validation, it also worked super well with forward selection. The model yielded a R^2 bar of 0.329, a R^2 cv of 0.330, AIC of -4122.434, and BIC of -4073.563 for the following features: volatile acidity, residual sugar, alcohol, density, pH, sulphates, free sulfur dioxides, fixed acidity, total sulfur dioxide, and chlorides.

Overall, both models from the two languages provided similar results with Python having slightly more accurate values. The 3L neural network with reLU activation function for either cross-validation and forward selection would make great models for this particular dataset. To view more, review the findings in the results excel sheet.